

Weekly Report(2018.11.19-2018.11.25)

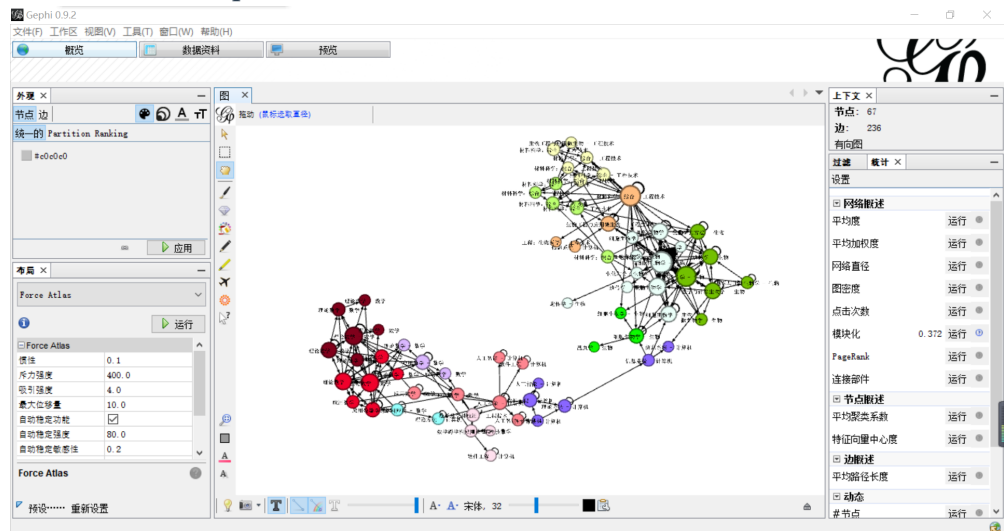
DONE

1. 全栈demo (D3.js + Node.js + MySQL) :

- 项目意义：在大二的健行创新基金中对微软学术数据（MAG）进行了数据可视化研究，使用基于海量文献数据中的论文引用关系进行期刊/会议的自动聚类划分。本项目将实践全栈流程，重构该学术网络分析系统。
- 使用的语言部件：
 - D3.js（前端，渲染期刊聚类情况的force-directed graph）
 - Node.js（后端，运行服务器，提取MySQL数据传输至D3）
 - MySQL（数据库，存储期刊聚类的原始数据，包括nodes集和edges集）

11.20 周二

- 1、原项目在完成数据挖掘、数据分析后得到存储nodes集和edges集的两个CSV文件，并使用Gephi的交互式界面制作聚类，选择Force Atlas布局：



使用gephi

- 2、在MySQL中创建存储nodes集和edges集的tables，并读入CSV文件；
- 3、在数据库中存储了期刊的关系矩阵数据如下，其中对于边的数据表去除了期刊与自身的引用关系：

```
delete from j2j_edge where j_source = j_target;
```

j_id	j_label	j_label2	j_name
1	计算机	理论方法	Information Fusion
2	工程技术	材料科学：综合	Nature Nanotechnology
3	生物	细胞生物学	Science Translational Medicine
4	数学	理论数学	Publications Mathematiques De L Ihes
5	数学	数学跨学科应用	Communications In Nonlinear Science...
6	工程技术	数学跨学科应用	Computer Methods In Applied Mecha...
7	计算机	理论方法	Iee Transactions On Neural Network...

nodes集

	j_source	j_target	j_edge_id	j_weight
▶	1	29	1	1
	1	50	2	1
	1	65	3	3
	2	36	5	1
	2	46	6	1
	3	25	8	1
	3	32	9	1
	3	37	10	2

edges集

11.21 周三

- 1、通过Node.js连接MySQL，并读取数据打包成Json格式，用于传输给D3.js：

127.0.0.1:8081/get_D3_j2j?user_name=wang&user_pwd=123456

QQ邮箱 网易邮箱 健行学院 教务系统 浙大软院 GitHub CPR 2018 McFate 翻译 VAG 127.0.0.1

```
[{"j_id":1,"j_label":"计算机","j_label2":"理论方法","j_name":"Information Fusion"}, {"j_id":2,"j_label":"工程技术","j_label2":"材料科学：综合",
{"j_id":3,"j_label":"生物","j_label2":"细胞生物学","j_name":"Science Translational Medicine"}, {"j_id":4,"j_label":"数学","j_label2":"理论",
{"j_id":5,"j_label":"数学","j_label2":"数学跨学科应用","j_name":"Communications In Nonlinear Science And Numerical Simulation"}, {"j_id":6,
用","j_name":"Computer Methods In Applied Mechanics And Engineering"}, {"j_id":7,"j_label":"计算机","j_label2":"理论方法","j_name":"IEEE Tr
{"j_id":8,"j_label":"生物","j_label2":"遗传学","j_name":"Genome Research"}, {"j_id":9,"j_label":"数学","j_label2":"理论数学","j_name":"Anne
术","j_label2":"材料科学：综合","j_name":"Nano Energy"}, {"j_id":11,"j_label":"生物","j_label2":"数学与计算生物学","j_name":"Bioinformatic
物理学","j_name":"Annual Review Of Biochemistry"}, {"j_id":13,"j_label":"生物","j_label2":"微生物学","j_name":"Nature Reviews Microbiology"},
合","j_name":"Materials Today"}, {"j_id":15,"j_label":"工程技术","j_label2":"数学跨学科应用","j_name":"Archives Of Computational Methods In
术","j_label2":"生物工程与应用微生物","j_name":"Biotechnology Advances"}, {"j_id":17,"j_label":"数学","j_label2":"理论数学","j_name":"Nonli
{"j_id":18,"j_label":"数学","j_label2":"应用数学","j_name":"Applied And Computational Harmonic Analysis"}, {"j_id":19,"j_label":"数学","j_l
Mathematical Society"}, {"j_id":20,"j_label":"生物","j_label2":"遗传学","j_name":"Trends In Ecology & Evolution"}, {"j_id":21,"j_label":"数
European Mathematical Society"}, {"j_id":22,"j_label":"工程技术","j_label2":"材料科学：综合","j_name":"Progress In Materials Science"}, {"j
学","j_name":"Briefings In Bioinformatics"}, {"j_id":24,"j_label":"生物","j_label2":"遗传学","j_name":"Annual Review Of Genetics"}, {"j_id":
Review Of Entomology"}, {"j_id":26,"j_label":"生物","j_label2":"细胞生物学","j_name":"Annual Review Of Cell And Developmental Biology"}, {"
```

传至前端的json数据

- 2、编写Node.js，运行服务器，使用路由，编写用户登录功能，登录成功后先能调用上述功能，传输Json数据：

127.0.0.1:8081/index

QQ邮箱 网易邮箱 健行学院 教务系统 浙大软院 GitHub CPR 2018 McFate

登录以查看D3期刊聚类图：

用户名：

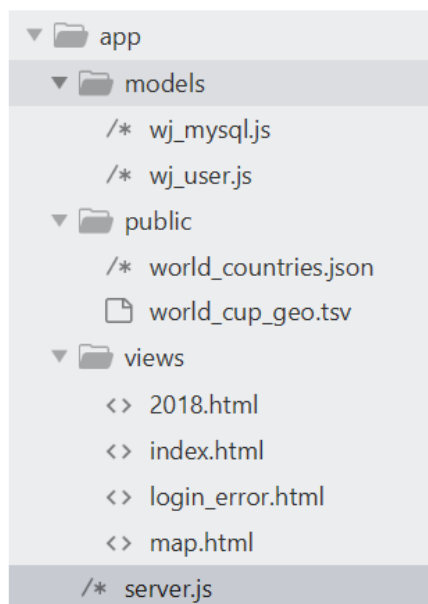
密码：

免登录查看D3样例：

登录功能

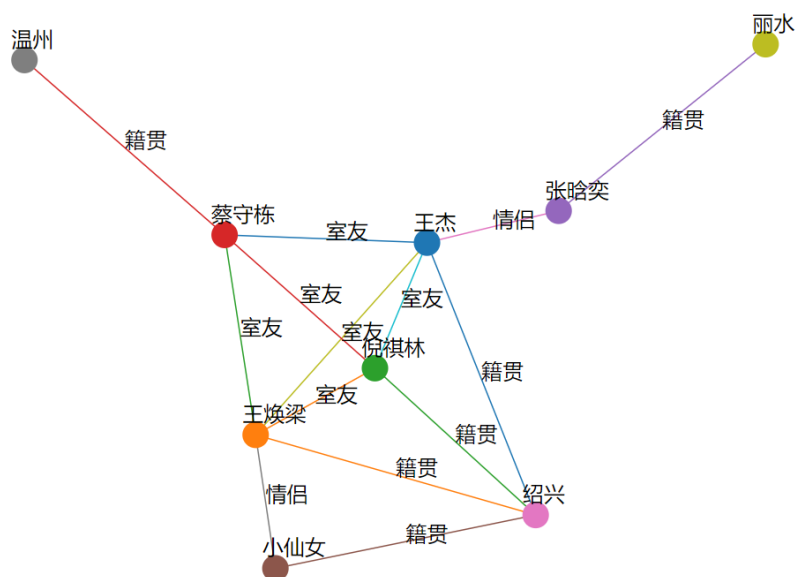
11.22 周四

- 1、使用类似MVC的设计模式思想整理了Node.js的项目结构，能够在服务器中运行之前完成的world cup样例：



项目结构

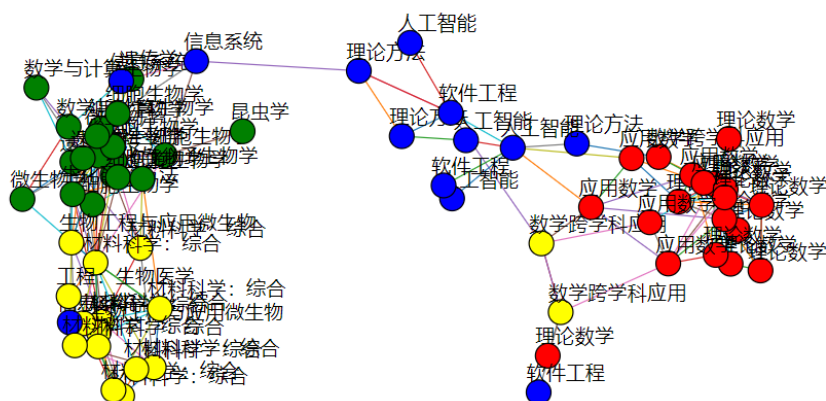
- 2、学习D3力导向图，实现demo：官网的example没注释看不懂，找了中文v5的博客学习，先实现了这个小demo：



同学关系和籍贯FDG图

11.23 周五

- 1、将期刊数据整理成了服务器上的json数据，更改了力导向图的代码，实现了读取服务器json文件展示期刊聚类功能：



- 2、存在的问题是，此时还不能将读取到的MySQL数据传送至前端Html网页。询问学姐后，需要后续学习 `$.ajax` 和 `promise` 进一步实现将后端的数据传输至前端。

小结

- 本周完成了全栈流程，能较好地掌握前端、后端的连接通信，同时实现了在服务器上运行D3demo。接下来要看TCP Tree论文和阅读Java代码了。

学习记录

学习日期	学习事项	学习时间
周一	回老家，休息	0h
周二	全栈学习	6h
周三	全栈学习	7h
周四	全栈学习	6h
周五	听组会，全栈学习，向学姐汇报，开始使用马克飞象	9h
周六	整理	5h
周日	报名，看论文TCP Tree	4h

PLAN

短期计划

- 看论文TCP Tree中文版5.2章,看对应Java代码并加上注释。
- 自学less、webpack、promise、\$.ajax、VS Code + Git等等。

中期计划

- 看论文TCP Tree对应Java代码并加上注释。

长期计划

- 跟紧VIS 2019的项目进度。
- 使用所学的网页工具实现自己的idea。

APPENDIX

- 在MySQL中创建存储nodes集和edges集的tables，并读入CSV文件：

```
1. load data infile 'C:\\mytools\\MySQL\\file\\node.csv'
2. into table j2j_node character set gb2312
3. fields terminated by ',' optionally enclosed by '"' escaped by '"'
```

```

4. lines terminated by '\r\n'
5. ('j_id', 'j_label2', 'j_label', 'j_name' );
6. load data infile 'C:\\mytools\\MySQL\\file\\edge.csv'
7. into table j2j_edge character set gb2312
8. fields terminated by ',' optionally enclosed by '"' escaped by '\"'
9. lines terminated by '\r\n'
10. ('j_source', 'j_target', 'j_weight');

```

2. 通过Node.js连接MySQL, 并读取数据打包成Json格式, 用于传输给D3.js:

```

1. exports.load_mysql = function(req, res){
2.     // 引入MySQL模块
3.     var mysql = require('mysql');
4.     // 新建连接对象, 输入用户密码数据库名称
5.     var connection = mysql.createConnection(
6.         {
7.             host : 'localhost',
8.             user : 'root',
9.             password: 'qweasd9095',
10.            port : '3306',
11.            database : 'web_user'
12.        });
13.    // 建立连接
14.    connection.connect();
15.    // 定义sql查询语句
16.    var sql = 'select * from j2j_node';
17.    // 在连接中发起query语句, 返回结果到数组rows中
18.    var data = {}; // 用于保存query返回的数据
19.    connection.query(sql, function(err, result) {
20.        if(err){
21.            console.log('语句执行发生错误 - ', err.message);
22.            return;
23.        }
24.        console.log('成功连接到数据库...\n数据传输中.....');
25.        //使用res的json参数返回数据
26.        res.json(result);
27.    });
28.    // 关闭连接
29.    connection.end();
30.    }

```

3. 编写Node.js, 运行服务器, 使用路由, 编写用户登录功能, 登录成功后先能调用上述功能, 传输Json数据:

```

1. // 用户登录后获取J2j界面
2. app.get('/get_D3_j2j', function (req, res) {
3.     //通过验证返回json数据
4.     if(wj_user.check_user(req, res)) {
5.         wj_mysql.load_mysql(req, res);

```

```
6.         }
7.         else {
8.             res.sendFile( __dirname + "/views/login_error.html" );
9.         }
10.    })
```